HOUSE PRICE PREDICTION USING MACHINE LEARNING

Phase- 3 Submission Document

Project Title : House Price Predictor

Phase 3 : Development part 1

Topic : Continue building the house price prediction model by feature engineering, model training, and evaluation.

HOUSE PRICE PREDICTION USING MACHINE LEARNING

Introduction :

House price prediction through machine learning is an essential and increasingly popular application of data science and artificial intelligence. It involves the development of predictive models that can estimate the selling price of residential properties based on a variety of features such as size, location, number of bedrooms, and numerous other attributes.

The journey begins with the acquisition and loading of a comprehensive dataset containing historical information about houses. Once the dataset is obtained, meticulous data preprocessing is crucial. This phase includes addressing missing data, encoding categorical variables, scaling numerical features, handling outliers, and selecting relevant attributes for analysis. These initial steps lay the foundation for training and evaluating machine learning models, which ultimately empower us to make informed and accurate predictions, aiding both buyers and sellers in the ever-evolving real estate market.

The process commences with the acquisition of a rich and diverse dataset, loaded into the analytical environment. This dataset may encompass a multitude of factors, including property characteristics and market indicators. Subsequently, data preprocessing becomes paramount as it involves cleaning, transforming, and enhancing the dataset.

DATA LOADING AND PREPROCESSING

Step 1: Load the Dataset

You first need a dataset that contains information about houses, including features like size, location, number of bedrooms, bathrooms, etc., and their corresponding sale prices. You can find such datasets on various platforms like Kaggle, UCI Machine Learning Repository, or real estate websites. Let's assume you have a CSV file named 'house_prices_dataset.csv'.

PYTHON CODE

```
import pandas as pd

# Load the dataset

data = pd.read_csv('house_prices_dataset.csv')
```

Step 2: Explore the Dataset

Before diving into machine learning, it's important to understand your dataset. Check its structure, data types, and some sample records.

PYTHON CODE

```
 # Display the first few rows of the dataset
```

```
print(data.head())

# Get summary statistics

print(data.describe())

# Check for missing values

print(data.isnull().sum())
```

Step 3: Data Preprocessing

Data preprocessing is a crucial step to ensure that your dataset is suitable for machine learning. This may include handling missing values, encoding categorical variables, and scaling numerical features.

PYTHON CODE

```
# Handle missing values (example: fill with the mean)

data.fillna(data.mean(), inplace=True)

# Encode categorical variables (example: one-hot encoding)

data = pd.get_dummies(data, columns=['categorical_feature'])

# Split the data into features (X) and the target variable (y)

X = data.drop('sale_price', axis=1)

y = data['sale_price']
```

Step 4: Split the Data

Divide your dataset into a training set and a testing set. This is essential for evaluating your machine learning model's performance.

PYTHON CODE

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Step 5: Choose and Train a Machine Learning Model

Select an appropriate machine learning model for regression, such as Linear Regression, Decision Trees, Random Forest, or Gradient Boosting. Train the chosen model using the training data.

PYTHON CODE

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()

model.fit(X_train, y_train)
```

Step 6: Evaluate the Model

Assess the model's performance using regression metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R2) on the test data.

PYTHON CODE

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

y_pred = model.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
```

mse = mean_squared_error(y_test, y_pred)

r2 = r2_score(y_test, y_pred)

print(f"Mean Absolute Error: {mae}")

print(f"Mean Squared Error: {mse}")

print(f"R-squared: {r2}")

Step 7: Make Predictions

You can now use your trained model to make predictions for new or unseen data.

PYTHON CODE

new_data = pd.DataFrame(...) # Input features for a new house

predicted_price = model.predict(new_data)

This is a basic outline of how to perform house price prediction using machine learning by loading a dataset. Depending on the complexity of your dataset and the desired model, you may need to fine-tune and optimize your model for better predictive performance.

PYTHON CODE FOR DATASET PREPROCESSING

We'll use the Pandas and Scikit-Learn libraries for loading and preprocessing the data

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

```
data = pd.read_csv('house_prices_dataset.csv')

data.fillna(data.mean(), inplace=True)

data = pd.get_dummies(data, columns=['categorical_feature'])

scaler = StandardScaler()

numerical_features = ['feature1', 'feature2', 'feature3']  # List of numerical feature names

data[numerical_features] = scaler.fit_transform(data[numerical_features])

new_data = pd.DataFrame(...)

predicted_prices = model.predict(new_data)
```

USED MODULES

House price prediction using machine learning typically involves the use of several Python libraries or modules to load, preprocess, train, and evaluate your models. Here are the key modules commonly used in the process.

1. Pandas:

Purpose: Data manipulation and analysis.

Usage: Load the dataset, explore and preprocess data, handle missing values, and perform feature engineering.

Example: import pandas as pd

2. NumPy:

Purpose: Numerical computing and data manipulation.

Usage: Work with arrays and numerical operations.

Example: import numpy as np

3. Scikit-Learn (sklearn):

Purpose: Machine learning tools and algorithms.

Usage: Choose, train, and evaluate machine learning models.

Example: import sklearn

4. Matplotlib:

Purpose: Data visualization.

Usage: Create plots and visualizations to understand data.

Example: import matplotlib.pyplot as plt

5. Seaborn:

Purpose: Data visualization (built on Matplotlib).

Usage: Create more attractive and informative statistical graphics.

Example: import seaborn as sns

6. Scipy:

Purpose: Scientific and technical computing.

Usage: Provides additional statistical functions for analysis.

Example: import scipy

7. Statsmodels:

Purpose: Statistical modeling and hypothesis testing.

Usage: Conduct statistical tests and model analysis.

Example: import statsmodels.api as sm

8. StandardScaler:

Purpose: Feature scaling.

Usage: Standardize or normalize numerical features for better model performance.

Example: from sklearn.preprocessing import StandardScaler

9. Train-Test Split (from sklearn.model_selection):

Purpose: Data splitting for model training and evaluation.

Usage: Split your dataset into training and testing subsets.

Example: from sklearn.model_selection import train_test_split

10. Regression Models (e.g., LinearRegression, RandomForestRegressor, XGBoostRegressor):

Purpose: Machine learning algorithms for regression tasks.

Usage: Select and train a regression model on your dataset.

Examples:

```
from sklearn.linear_model import LinearRegressionfrom
sklearn.ensemble import RandomForestRegressor

from xgboost import XGBRegressor
```

11. Metrics (e.g., mean_absolute_error, mean_squared_error, r2_score from sklearn.metrics):

Purpose: Model evaluation metrics.

Usage: Assess model performance with appropriate metrics.

Examples:

```
from sklearn.metrics import mean_absolute_error,
mean_squared_error, r2_score
```

CONCLUSION

In conclusion, house price prediction using machine learning, coupled with dataset loading, represents a powerful and data-driven approach to understanding the complexities of the real estate market. By leveraging historical data, we can develop predictive models that assist both buyers and sellers in making informed

decisions. The process begins with the acquisition and careful loading of a well-structured dataset, followed by rigorous data preprocessing to ensure data quality and consistency.

Through the application of various machine learning algorithms and regression models, we can effectively predict house prices, taking into account a multitude of influential factors. This predictive capability has the potential to revolutionize the real estate industry, offering valuable insights, enhancing transparency, and enabling stakeholders to navigate the market with confidence.

Nonetheless, it is essential to continually refine and optimize these models to keep pace with the ever-evolving dynamics of the real estate sector, ensuring accurate and reliable predictions.